

通過テスト 1 2 解答

12 入力と出力

1. C, D, E, H

【解説】323 頁のクラス一覧表から、`BufferedWriter` のコンストラクタは `Writer` 型の参照を引数に取る。 `Writer` のサブクラスでもよい。 `A` は `File` 型、 `B` は `String` 型の引数なので正しくない。 `C` の `FileWriter` は `Writer` のサブクラスなので正解。同様に `PrintWriter` のコンストラクタ引数は `File`, `String`, `OutputStream`, `Writer` である。 `D` は `String` 型、 `E` は `File` 型なので正しい。 `BufferedReader` のコンストラクタ引数は `Reader` 型かそのサブクラスなので `F` は正しくない。 また、 `G` では `BufferedReader` のコンストラクタ引数は `Reader` 型だが、 `Reader` にはファイル名などの文字列を引数にとるコンストラクタがないので `G` も誤りである。 `H` の `FileReader` は `Reader` のサブクラスなので正しい。

2.

```
package e2;
import java.io.*;
import lib.Input;
public class FileTest {
    public static void main(String[] args) {
        String path = Input.getString("ファイルパス");
        try {
            PrintWriter out =
                new PrintWriter(new BufferedWriter(new FileWriter(path)));
            String s;
            // Enterだけをタイプすると null が入力される
            while((s=Input.getString())!=null){
                out.println(s);
            }
            out.close();
        } catch (IOException e) { e.printStackTrace(); }
        try {
            BufferedReader in = new BufferedReader(new FileReader(path));
            String s;
            while((s=in.readLine())!=null){
                System.out.println(s);
            }
            in.close();
        } catch (IOException e) { e.printStackTrace(); }
    }
}
```

【解説】SPD では `try-catch` を書いていないが、ファイル入出力ではチェック例外が発生するので `try-catch` で囲む必要があることに注意。

3. E

【解説】**File** クラスのオブジェクトを作成しても例外は発生しない。勘違いしやすいので注意する。問題のように **try-catch** で囲んでしまうと到達不可能な **catch** ブロック（例外が発生しないので）というコンパイルエラーになる。

4. D

【解説】**main** メソッドに“**throws Exception**”が付いているので、**try-catch** 文は不要であることに注意する。このため、発生した例外は **JVM (Java Virtual Machine)** で受け取られ、例外が発生するとプログラムの実行は停止する。

存在しないファイルについても **File** クラスのオブジェクトを作成できる。しかし、この問題では **BufferedReader** クラスのオブジェクトを作成する際にファイルが見つからず例外が発生する。つまり、実行時に **FileNotFoundException** が発生する。

5. D

【解説】**n1** はクラス変数、**n2** は **Serializable** インタフェースを実装していないスーパークラスのフィールド変数、**n3** は **transient** 指定されたフィールド変数、**n4** は普通のフィールド変数である。直列化復元時の値は、**n1** は直列化、直列化復元とは無関係なので設定した値 **10** がそのまま残っている。**n2** は直列化復元でスーパークラスのコンストラクタにより初期化されるので **0** になる。**n3** は直列化の対象にならないので変数の型の初期化規定値（**=0**）が充てられる。**n4** は正常に直列化時の値が復元され **40** になる。